Chaining Event Spans for Temporal Relation Grounding

Jongho Kim[♠♣], Dohyeon Lee[♠], Minsoo Kim[♠♣], Seung-won Hwang[♠♣*]
[♠]Seoul National University
[♣] Interdisciplinary Program in Artificial Intelligence, Seoul National University
{jongh97, waylight3, minsoo9574, seungwonh}@snu.ac.kr

Abstract

Accurately understanding temporal relations between events is a critical building block of diverse tasks, such as temporal reading comprehension (TRC) and relation extraction (TRE). For example in TRC, we need to understand the temporal semantic differences between the following two questions that are lexically near-identical: "What finished right before the decision?" or "What finished right after the decision?". To discern the two questions, existing solutions have relied on answer overlaps as a proxy label to contrast similar and dissimilar questions. However, we claim that answer overlap can lead to unreliable results, due to spurious overlaps of two dissimilar questions with coincidentally identical answers. To address the issue, we propose a novel approach that elicits proper reasoning behaviors through a module for predicting time spans of events. We introduce the Timeline Reasoning Network (TRN) operating in a two-step inductive reasoning process: In the first step model initially answers each question with semantic and syntactic information. The next step chains multiple questions on the same event to predict a timeline, which is then used to ground the answers. Results on the TORQUE and TB-dense, TRC and TRE tasks respectively, demonstrate that TRN outperforms previous methods by effectively resolving the spurious overlaps using the predicted timeline [1].

1 Introduction

Understanding temporal relations is a challenging yet underexplored area in natural language processing (Ning et al., 2020; Chen et al., 2021; Zhou et al., 2019). This challenge persists despite the prevalence of Large Language Models (LLMs) (Chan et al., 2023; Fang et al., 2023), whose training processes lack grounding in timeline evidence. One example task requiring such evidence is temporal reading comprehension (TRC), which requires to distinguish the temporal semantic difference between *"what **finished right before** the decision?"* and *"what **finished right after** the decision?"*.

To distinguish the two questions, existing solution for TRC (Shang et al., 2021) relies on overlaps between related questions as a weak supervision to ground the semantics of temporal relations. For example, in Figure 1, if we let the question's target event as X, $Q1$ *"what had **started before** X"* and $Q2$ *"what **happened before** X"* have similar semantics "***before***". Subsequently, the two share the overlapping answer *"sent"*. On the other hand, the temporal semantics of $Q2$ and $Q3$ *"what **happened while** X"* are different. So $Q2$ does not have any common answer with $Q3$. By using answer overlaps as a proxy label, existing work proposes a contrastive objective. It aims to pull the temporal relations in $Q1$ and $Q2$ closer together while broadening the distinction between $Q2$ and $Q3$. This method performs comparably with or outperforms baselines requiring stronger but expensive human-annotations (Han et al., 2021; Huang et al., 2022), as shown in Subsection 4.4.

However, as illustrated in Figure 2, we argue that contrasting the evidence from answer overlaps misguide timeline as **point-wise** manner, leading to "spurious overlap". Questions $Q3$ and $Q4$ *"What happened while X"* and *"What probably ended after X"*, are temporally distinct but share answers "taken" and "bearing". In such cases, the point-wise timeline may fail to properly reason about the temporal meanings of the two questions. The timeline mistakenly pulls $Q3$ and $Q4$ closer, making the model insufficient to differentiate the complex temporal questions. The point-wise representation misses the timeline's inherent **span-based** nature.

In this work, we focus on overcoming the limi-

*Corresponding author.
[1]Codes are available here: https://github.com/JonghoKimSNU/Temporal-Reasoning-Network/tree/main

Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 1689–1700

[e1] Aircraft have taken off from the United States, [e2] bearing medical supplies. A rescue team, [e3] previously sent to the bombed-out federal building in Oklahoma City, [e4] was en route to Nairobi.

Question (question group of [e4])	Answer
Q1. What had started before the team was en route to Nairobi?	**A. taken, bearing, sent**
Q2. What happened before the team was en route to Nairobi?	**A. sent**
Q3. What happened while the team was en route to Nairobi?	**A. taken, bearing**
Q4. What probably ended after the team was en route to Nairobi?	**A. taken, bearing**

Figure 1: Example of passage and question grouped by the same event ('the team was en route") in temporal reading comprehension. Events are highlighted in color and temporal relations in the questions are in red.

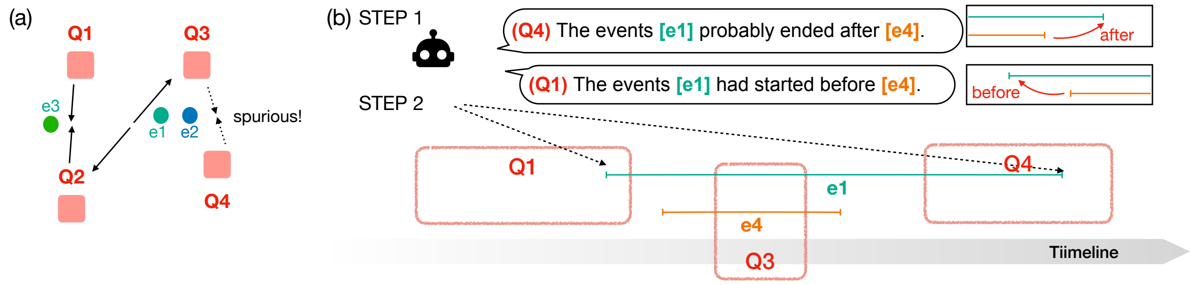

Figure 2: The illustration of (a) point-wise timeline grounding and (b) span-based one. (a) The model brings similar relations closer and pushes dissimilar ones apart, overlooking spurious overlap. (b) The speech bubbles in Step 1 describe the temporal evidence from each question-answer pair. The arrows in Step 2 describe the relative span prediction. It chains evidences about the timeline and mitigates spurious overlap.

tations of point-wise event representation through span-based representations of time. The key is utilizing the concept of time spans with notions of start and end points to supervise the complex temporal relationships between events. For instance, the timeline in Figure 2(b) can separate $Q3$ and $Q4$ and distinguish between *"happened while"* and *"probably ended after"*, which are illustrated as disjoint boxes. The overlap of the events is because the events span throughout the timeline, not because the questions are similar. Despite its importance, previous work does not consider such a timeline due to the limited supervision in most scenarios.

We propose an advanced solution that elicits inductive reasoning behavior from a model grounded in predicted event spans. Inductive reasoning in the context of temporal relation understanding is the process of extracting relations from individuals for deducing a whole, with the key purpose to acquire relative spans of events centered around a specific event. First, the model answers each temporal relation question in a "question group", a set of questions about the same event (e.g., $e4$). As illustrated in speech bubbles in Figure 2(b), the question-answer pairs can be understood as part of the evidence about the timeline, such as 'when event $e1$ occurred relative to the event $e4$'. Second, the model chains multiple temporal evidence within the same question group. This chained information forms a predicted timeline. For example, the speech bubbles in Figure 2(b) collectively illustrate the start and end points of event $e1$. Supervised by the predicted timeline, events that span a long time period can be identified, allowing us to discount attention to events with spurious overlaps. This process mitigates the spurious overlap without expensive human supervision.

Our model, Timeline Reasoning Network (TRN), equips the two-step inductive reasoning outlined as follows: An Evidence Extraction step aims to answer a specific question by extracting semantic and syntactic information with a pre-trained language model (PLM) and graph network. An Evidence Chaining step collectively predicts a timeline, using the novel attention module to chain multiple question-answer pairs. With the resulting timeline, the model grounds its answers consistently enhancing overall prediction accuracy.

We evaluate TRN on TORQUE and TB-Dense, a TRC and TRE task respectively. We achieve state-of-the-art performance on the public leaderboard of TORQUE [2]. We quantitatively and qualitatively analyze TRN's effectiveness in dealing with spurious overlaps, which is measured by our new proposed "passage level consistency" metric. Lastly, we confirm its generalizability on TB-Dense. Our main contributions are three-fold:

- We point out the spurious overlap issue in temporal relations, which arises from point-wise timeline grounding.

- We propose the inductive solution that chains

[2] https://leaderboard.allenai.org/torque/submissions/public

evidence for the timeline in a span-based approach.

- Our novel framework, TRN, outperforms other approaches by effectively capturing temporal relations of events.

2 Related Work

We overview state-of-the-art works on temporal relation understanding and graph networks.

Temporal relation understanding Temporal relation understanding remains a challenging task even for large language models (LLMs) (Chan et al., 2023). This includes task types such as TRE and TRC. TRE tasks (Cassidy et al., 2014; Ning et al., 2018) are to categorize the temporal order into pre-defined categories. MATRES (Ning et al., 2018) groups the temporal relations into 4 categories: *Before/After/Simultaneous/Vague*. TB-Dense (Cassidy et al., 2014) considers 2 more classes, *Includes* and *Is Included*. Our proposed approach can benefit these tasks as we discuss in Section 5.

Meanwhile, our main task is the TRC task TORQUE (Ning et al., 2020), requiring a temporal ordering in question form to reflect the real-world diversity of temporal relations. Previous approaches to the TRC task include continuous pre-training (Han et al., 2021) and question decomposition methods (Huang et al., 2022; Shang et al., 2021). ECONET (Han et al., 2021) continually pre-trains the model to inject the knowledge of temporal orders. Question decomposition approaches (Huang et al., 2022; Shang et al., 2021) divide the question into the event part and temporal relation expression part to better capture the complex semantics. All of the above use contrastive methods to understand different temporal relations, either by contrasting relations with human annotations (Han et al., 2021; Huang et al., 2022) or annotated answers (Shang et al., 2021). However, the former can be costly or imprecise, while the latter may rely on spurious problems. Our distinction is the best of the two: no costly human annotation while avoiding spurious overlaps using span-based inductive reasoning.

Graph networks Graph Networks (Kipf and Welling, 2016; Velickovic et al., 2017) learn features through message passing on graph structures. These networks have demonstrated their effectiveness in tasks requiring complex reasoning skills, such as numerical reasoning (Ran et al., 2019; Chen et al., 2020) and logical reasoning (Huang et al., 2021). Graph networks also have been applied to TRE (Cheng and Miyao, 2017; Mathur et al., 2021; Zhang et al., 2022), though their effectiveness in TRC has not been investigated.

3 Proposed Method

We formulate predicting answers for a query Q as a binary classification for every word p in the given passage P, determining whether it is an answer event to Q [3].

Our approach is to solve the task with the two steps of inductive reasoning. The core of inductive reasoning is inferring the whole picture from individual evidence. To transform the conventional function used in reading comprehension into the inductive form, we modify the function to consider answers to multiple questions together. The conventional one is denoted as $\hat{A}_i = f(Q_i, P; \theta)$, where we answer ($\hat{A}_i$) the i-th question (Q_i) in the passage with model θ. For inductive reasoning, the function is modified as:

$$\begin{aligned} \hat{A}_i^{induced} &= f(Q_i, P, \hat{A}^*; \theta), \\ \text{where } \hat{A}^* &= \{\hat{A}_i\}_{i=1}^{l} \end{aligned} \tag{1}$$

l is the number of questions, and $\hat{A}^*$ is the set of model predictions for multiple questions.

The overview of our model is in Figure 3. We first extract each answer ($\hat{A}_i$) as individual evidence in the Evidence Extraction step (Subsection 3.1), represented as the output squares in (a). The inductive reasoning is elicited in the Evidence Chaining step (Subsection 3.2). We chain the related question-answers ($\hat{A}^*$) depicted as paths of blue and red, marked with a dark background, and utilize them in (b).

3.1 Evidence Extraction Step

The evidence extraction step aims to extract timeline evidence by answering each question. We utilize both semantic information from PLM and syntactic information from the graph network. First, PLM encodes the question-passage pairs to get the contextual representation for each token. It takes the concatenated sequence of pair as input

[3] To facilitate a fair comparison with the available baselines in Section 4, we also adopted the practice of using the first token as a word if a word is split into multiple tokens.

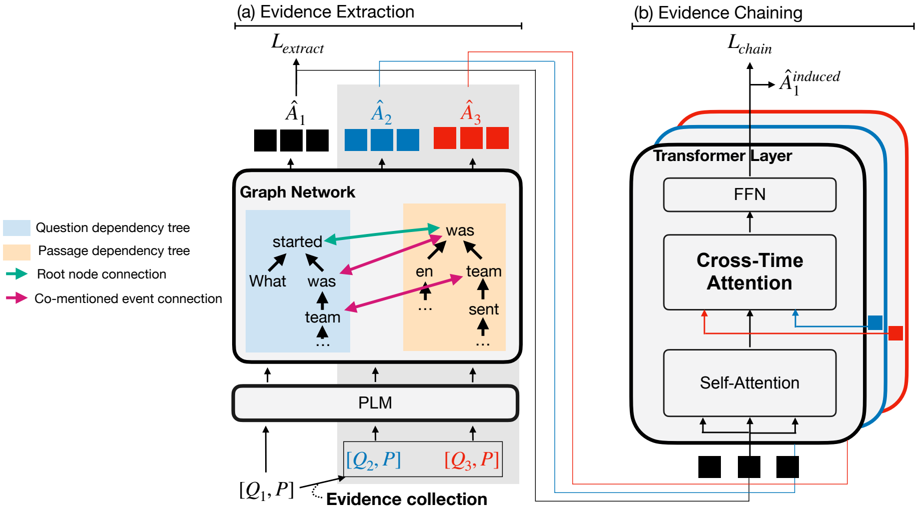

Figure 3: Overview of TRN. (a) The Evidence Extraction Step answers each question with semantic (PLM) and syntactic (Graph Network) features. The example in the graph is from $Q1$ in Figure 1. (b) The Evidence Chaining Step collects the related answers in the evidence collection stage and chains them through the cross-time attention module.

$[Q, P]$ and outputs the vector representation $[Q^v, P^v]$, where each token is q^v and p^v.

After that, we build a syntax-aware graph neural network that captures word-to-word dependency, which is an effective strategy for temporal reasoning (Cheng and Miyao, 2017; Mathur et al., 2021; Zhang et al., 2022). Diverging from previous works mainly focused on temporal relations within passages and neglected questions, our formulation highlights the need to comprehend both. As the graph in Figure 3(a), we construct dependency tree graphs for both the question and passage, connecting root nodes and co-mentioned event words bidirectionally to facilitate the information exchange. Here event words refer to nouns and verbs.

Next, we followed the graph reasoning step used in reading comprehension (Ran et al., 2019) that categorizes the connections of nodes into 4 types: (1) question-question (qq) (2) passage-passage (pp) (3) passage-question (pq) (4) question-passage (qp). Each node in the graph is the corresponding word in question and passage. The pipeline consists of the following steps:

$$[\bar{Q}, \bar{P}] = W^m[Q^v, P^v] + b^m \tag{2}$$

$$\alpha_i = sigmoid(W^v\bar{v}_i + b^v) \tag{3}$$

$$\tilde{v}_i = \frac{1}{|N_i|}\left(\sum^{j\in N_i} \alpha_j W^{r_{ji}}\bar{v}[j]\right) \tag{4}$$

$$v'_i = ReLU(W_i^u\bar{v}_i + \tilde{v}_i) + b^u \tag{5}$$

(a) Projection: The vector outputs of the PLM pass through the projection layer W^m for node initialization (Eq. 2). (b) Node Relevance: We compute the weight α_i for each node $\bar{v}_i$ with the sigmoid function to determine the relevant nodes for answering temporal ordering questions (Eq. 3). Here, nodes $\bar{v}$ consist of $\bar{q}$ and $\bar{p}$, each corresponding to the nodes from the question and passage. (c) Message Propagation: The adjacency matrix $W^{r_{ji}}$ guides the message passing between nodes of different types (Eq. 4), where $r_{ji} \in \{pp, pq, qp, qq\}$ and N_i is the neighbor nodes of $\bar{v}_i$. (d) Node Update: The message representations are added to the corresponding nodes, and a non-linear activation function (ReLU) is applied to update the node representations (Eq. 5).

We iterate the steps (b), (c), and (d) for T times. Finally, the representation from PLM, P^v, is added and normalized to obtain the answer representations $\hat{A}_i$ in Eq. 1, with individual word representations $\hat{a}_i$.

3.2 Evidence Chaining Step

Our second and primary objective is to inductively reason with the group of questions and ground the answers with it. A key motivation of the reasoning comes from the observation that chaining answers to questions about the same event serves as the relative timeline. Each prediction can be interpreted as temporal evidence like 'when one event occurred relative to the asked event'. The pieces of evidence are then chained with the attention module to create the relative time span of passage events, helping the model ground its predictions.

The evidence chaining step is built for such reasoning, whose process is further divided into two stages: evidence collection and timeline acquisition.

Evidence collection We first collect the question group, defined as questions that pertain to the same target event. Blue and red questions in Figure 3 correspond to the group. Task designs may provide the grouping for evaluation metrics (Ning et al., 2020) (Subsection 4.3) or simple rules can be applied for the grouping (Subsection 5.2).

The questions are collectively encoded through the evidence extraction step and the output representations of them are collected. If the model wants to answer the first question $[Q_1, P]$ in the question group, the other questions $[Q_i, P]_{i=2}^{l}$ are encoded together to produce $[\hat{A}_i]_{i=2}^{l}$, then stacked with the original one to make $\{\hat{A}_i\}_{i=1}^{l}$, which corresponds to $\hat{A}^*$ in Eq. 1.

Timeline acquisition We need to build the timeline from the collected evidence to ensure the model's original answer is consistently grounded in such a timeline.

We achieve this through a novel transformer layer with our key component "cross-time attention" module. Let the attention module $Attention(Q, K, V)$ (Vaswani et al., 2017). Conventional self-attention attends to tokens within a single data sequence-wise, represented by $Attention(p_{ki}, p_{kj}, p_{kj})$, where k is the data index and i, j are both equal or less than the sequence length. In contrast, our novel cross-time attention operates data-wise, gathering information from multiple data that were previously overlooked (Vaswani et al., 2017). Each passage token p_k attends to the same positioned token from related data. The equation of cross-time attention is:

$$CrossTimeAttention = Attention(p_{ik}, p_{jk}, p_{jk}) \quad (6)$$

where $i, j \leq l$ and k is token index. We insert cross-time attention between the self-attention and feed-forward network (FFN) in the transformer layer.

In the evidence chaining step, the answer $(\hat{a}_i)$ for the event (p) in i-th related question conveys evidence of when event p occurred relative to the event in question. Therefore, if the cross-time attention chains the pieces of temporal evidence of the event together, $\{\hat{a}_i\}_{i=1}^{l}$, it results in the time span of the event p. The resulting time spans for events allow the model to refine the answer by collectively leveraging them as ground evidence.

We enhance the model's reasoning behavior in temporal relation understanding through iterative application of our transformer layer T' times.

3.3 Training and Answer Prediction

At each step, the last output is fed to the one-layered perceptron head to get the prediction of whether the token is an answer to the question or not. During the training phase, the final loss is the mean of extraction and chaining step losses, rewarding output from both steps. The answer prediction loss from the first step, $L_{extract}$, guides the evidence from individual questions. The second step's loss, L_{chain}, guides the model to inductively correct the answer with the predicted timeline. During the inference phase, our final logits, $\hat{A}_i^{induced}$, are the predictions of the evidence chaining step.

4 Experiment

4.1 Dataset and Evaluation Metrics

We evaluate our proposed model on TORQUE dataset (Ning et al., 2020), which is a temporal reading comprehension dataset. It has 3.2k passages and 21.2k user-provided questions. Each instance has a question asking the temporal relationships between events described in a passage of text. TORQUE's annotation provides groups of questions, where one group consists of questions that were created by modifying the temporal nuance of an original seed question that dramatically changes the answers. We use the official split [4] and evaluation metrics, which include Macro F1, exact-match (EM), and consistency (C) as evaluation metrics. C (consistency) is the percentage of question groups for which a model's predictions have $F1 \geq 80\%$ for all questions in a group.

4.2 Baselines

We compare our model against several baselines, including PLMs and models that use contrastive methods to teach the model temporal relations. Specifically, **OTR-QA** (Shang et al., 2021) reformulates the TORQUE task as open temporal relation extraction and uses answer overlap to weakly supervise temporal relations. As they target TORQUE without any external supervision like our method, they are our main baseline. We further compare our model with those that use human-annotated temporal dictionaries. **ECONET** (Han

[4] https://github.com/qiangning/TORQUE-dataset/tree/main

Models	wo external sup.	significance	Dev			Test		
			F1	EM	C	F1	EM	C
RoBERTa-large	-	-	75.7	50.4	36	75.2	51.1	34.5
DeBERTa-large-v3	-	-	76.4	50.8	36.2	76.3	52.2	37.3
ECONET	X	O	76.9	52.2	37.7	76.3	52.0	37.0
UBA	X	X	77.5	52.2	37.5	76.1	51.0	**38.1**
OTR-QA	O	X	77.1	51.6	**40.6**	76.3	52.6	37.1
TRN (RoBERTa-large)	O	O	**77.6**rd	**53.6**^{rde}	40.3^{rde}	**76.9**^{rde}	**52.8**^{rde}	**38.1**^{r}

Table 1: Comparison between TRN and baselines on TORQUE dataset. We marked the models that (1) are trained without external supervision (2) have performed significance test on the test set. Superscripts represent significant improvements compared to RoBERTa(r), DeBERTa(d) and ECONET(e). The best performance is denoted in bold.

et al., 2021) is a continual pre-training approach with adversarial training that aims to equip models with knowledge about temporal relations. They use the external corpus and compile a dictionary of 40 common temporal expressions. **UBA** (Huang et al., 2022) employ the attention-based question decomposition to understand fine-grained questions. They also utilize a dictionary of temporal expressions as additional supervision, to capture the distinctions in temporal relationships. **RoBERTa-large** (Liu et al., 2019) is a baseline PLM provided together with the TORQUE dataset and the previous SOTAs are based on. In addition, we evaluate the score of **DeBERTa-v3-large (He et al., 2022)**, which is known as the state-of-the-art PLM on a wide range of natural language understanding tasks.

We don't regard recent LLMs as our main baseline due to their subpar performance in temporal relation understanding (Chan et al., 2023). Additional evidence supporting this assessment is presented in our extended evaluation of ChatGPT in Appendix A.

4.3 Experimental Settings

We search for optimized hyperparameters in our model. T and T' are set between {2, 3} for the graph iteration step and for the evidence chaining step respectively. Each transformer layer in the evidence chaining step has 8 attention heads with a hidden size of 1024, and FFN layers in the attention module have dimensions between {1024, 2048}. The question group is annotated for the C metric in TORQUE. During the fine-tuning, the gradient accumulation step is set to 1, dropout ratio is set to 0.2 and other settings are identical with Ning et al. (2020). Spacy (Honnibal et al., 2020) is used for graph construction. We use the PyTorch 1.11 library, and a NVIDIA GeForce RTX 3090 GPU with 42 average minutes to run an epoch.

For the performance report, we report the average score on the dev set and the best score on the test set to make a fair comparison with the baselines. This is because OTR-QA only reports the best single model results for all sets, and UBA reports single model results on the test set.

For the significance test, we conducted paired t-tests ($p < 0.05$) only with PLMs and ECONET. It was due to the lack of reproducibility and significance test on the test set for OTR-QA and UBA.

4.4 Experimental Results

Table 1 compares our approach to the baseline methods. The baseline performances are provided by previous works (Ning et al., 2020; Han et al., 2021; Shang et al., 2021; Huang et al., 2022). The results show that TRN outperforms all compared baselines on both splits of TORQUE. TRN even surpasses ECONET and UBA which use a human-annotated dictionary of temporal expressions. Moreover, we found that while DeBERTa-v3-large shows a comparable score with OTR-QA, TRN significantly beats both DeBERTa-v3-large and OTR-QA. Such results indicate our approach shows notable benefits over existing methods. One exception is the consistency score (C) of OTR-QA on the dev set. But we note that TRN outperforms it in F1 and EM and generalizes better to the test set, indicated by a much smaller dev-test gap in C (3.5 for OTR-QA vs 2.2 for TRN). On the test set, TRN significantly outperforms all the baselines, achieving SOTA results on the TORQUE leaderboard.

4.5 PLM variants

Table 2 displays the results for PLM encoder variants. First, we implement our method on DeBERTa-v3-large (He et al., 2021) and observe that with the addition of TRN, it achieves the best test scores across all metrics. It demonstrates the effectiveness and generalizability of our method even with other

Models	Dev			Test		
	F1	EM	C	F1	EM	C
DeBERTa-large-v3						
Naive	76.4	50.8	36.2	76.3	52.2	37.3
TRN	77.5	51.7	39.1	**77.3**	**53.5**	**38.5**
BERT-large						
Naive	72.8	46.0	30.7	71.9	45.9	29.1
Current SOTA	73.5‡	46.5‡	31.8‡	**72.6**‡	45.1‡	**30.1**‡
TRN	73.1	47.2	32.6	72.3	**46.5**	29.8
RoBERTa-base						
Naive	72.2	44.5	28.7	72.6	45.7	29.9
Current SOTA	75.2†	49.2†	36.1†	73.5‡	**47.1**†	**32.7**†
TRN	73.8	48.9	34.7	**73.7**	**47.1**	32.3

Table 2: Comparison with PLM variants. Naive results of BERT-large and Roberta-base are from TORQUE (Ning et al., 2020) and DeBERTa-large from our own implementation. Current SOTA results are from OTR-QA (Shang et al., 2021)†, UBA (Huang et al., 2022)‡.

Models	F1	EM	C
TRN	**77.6**	**53.6**	**40.3**
(d) TRN - Self-Attention	77.4	52.2	38.9
(c) TRN - Cross-Time Attention	76.3	51.4	38.6
(b) TRN - Evidence Chaining Step	76.0	51.9	38.1
(a) TRN - G_{syn}	76.1	50.9	37.3

Table 3: Ablation study on the dev set of TORQUE. Results are based on RoBERTa-large. The best performance is denoted in bold.

PLM variants. Our method is also shown to be generalizable to the BERT model, and its performance is comparable to other previous methods. Lastly, when using the RoBERTa-base model, our results are again comparable to other baselines and surpass them in terms of F1 score, highlighting the scalability of TRN.

4.6 Ablation Study

To validate the effectiveness of each model component, we conduct an ablation study on the dev set and report the results in Table 3. In (a) we remove the syntactic graph network component G_{syn} in the evidence extraction step and find the performance decreases significantly. This suggests that syntactic graph reasoning helps the downstream process of inductive reasoning by creating passage token representations more effectively. For the evidence chaining step, we first remove (b) the whole layer, (c) the cross-time attention layer, and (d) the self-attention layer. The performance drops significantly with (b), indicating the importance of the evidence chaining step. Comparison between (c) and (d) indicates that the event chaining step helps performance gain by virtue of cross-time attention. It is the leading part of our reasoning elicitation by attending over the predicted timeline. Meanwhile, (d) removing the simple stack of the transformer's self-attention part has the least impact on the performance.

5 Discussion

While we empirically validated the effectiveness of TRN, its implication and generalizability can be further clarified by the following discussion questions:

- Q1: Does TRN mitigate spurious overlaps?
- Q2: Does TRN generalize to another task?

5.1 Q1: Mitigating spurious overlaps

As we have claimed comprehension of a span-based timeline works as a key constraint to avoid spurious overlaps, we first address the question of whether the performance gain of TRN can be attributed to a better comprehension of the timeline in the passage.

To quantitatively measure whether TRN understands passage timelines, we adopt a passage-level consistency score C_p. In TORQUE, each passage contains multiple question groups and each question group has questions asking about the same event. The original evaluation metric C in Subsection 4.1 measures consistency at a specific event or time point within the passage, by considering answer consistency in one question group. On the other hand, C_p assesses the answer consistency across questions targeting different events by measuring the overall consistency of answers across multiple question groups within the same passage. We define C_p as the percentage of passages for which a model's predictions have $F1 \geq 80\%$ for all questions in a passage [5].

Through evaluating the consistency of answers across different time points corresponding to each target event, the C_p score provides insights into the model's understanding of the time spans of events. Therefore, if a model understands the passage timeline, its answers will be internally consistent with respect to the questions with different target events, which C_p quantifies. We compare TRN with the model equipped with contrastive learning (*CL*), which is implemented following OTR-QA's contrastive loss (Shang et al., 2021).

Table 4 shows that C_p of TRN is significantly higher than that of *CL*. To isolate the effect of the

[5] We use the threshold of $F1 \geq 80\%$ for all the questions in the passage following the convention of Gardner et al. (2020); Ning et al. (2020).

Models	F1	EM	C	C_p
(a) Extract + Chain (TRN)	**77.6**	**53.6**	**40.3**	**11.7**
(b) Chain	76.1	50.9	37.3	10.3
(c) *CL*	75.8	51.7	36.8	8.3

Table 4: Comparison of *CL* and TDN on the dev set of TORQUE. The best performance is denoted in bold.

chaining step where the model reasons to predict the timeline, we also present ablated results removing the extraction step. We observe that even without the evidence extraction TRN outperforms *CL*, which indicates that the improved understanding of timeline plays a critical role in mitigating spurious overlap and thereby achieving performance gains [6].

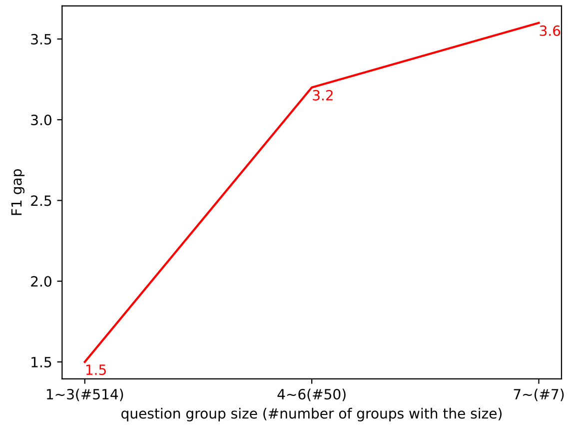

Figure 4: Plot of the relationship between the question group size and F1 score gap. X-axis is the group size, binned into groups of 3. The number of groups in each bin is denoted in brackets. Y-axis is the gap between the average F1 score of TRN and *CL*, in percentage.

Figure 4 groups F1 gains, by related question group sizes, from which the gap from *CL* widens as the size grows. It is coherent with our hypothesis that TRN gains effectiveness by the timeline information predicted from multiple related questions, which would be more effective for a larger question group size. Moreover, our method persistently outperforms contrastive loss, even with a small question group size with a margin of $1.5pp$.

Lastly, as qualitative observations, Figure 5 in Appendix B compares answers from TRN with *CL*: *CL* fails to clearly distinguish the semantic difference between Q1 and Q2, while our reasoning for the timeline avoids such mistakes. TRN is aware that "exploded" occurred before the tour ($Q3$), and not after the tour ($Q2$), so it cannot be during the same time as the tour ($Q4$). while *CL* fails. In addition, TRN finds the unmentioned events (e.g. "arrested" in $Q1$) and puts them in the right place on the timeline.

Models	Dev	Test
RoBERTa-large	58.9($\pm$2.1)	63.4($\pm$2.3)
ECONET	60.8($\pm$0.6)	64.8($\pm$1.4)
TRN	**62.5**($\pm$1.4)	**65.8**($\pm$0.6)

Table 5: Micro-F1 scores on the TB-Dense dataset. The best performance on the test set is denoted in bold.

5.2 Q2: Generalization

To investigate whether our proposed approach generalizes to other temporal relation understanding task, we evaluate our method on TB-Dense (Cassidy et al., 2014), which is a public benchmark for temporal relation extraction (TRE).

For TB-Dense, when the passage and two event points in the passage are given, the model must classify the relations between events into one of 6 types. As the explicit question is not provided in TB-Dense, we treat two event points as a question and group the questions in the dataset with a simple rule as follows: In the evidence extraction step, we prepend two events, $e1$, $e2$, to the passage P, and the model input is "$[CLS] + e1 + e2 + [SEP] + P + [SEP]$". In the evidence chaining step, we manually gather questions that are asked on the same first event within the same part of the passage, which can be easily identified by basic lexical matching. We use this gathering to construct the question group and predict the timeline. We implement our method based on the publicly available source code of ECONET (Han et al., 2021) [7]. Hyperparameters for fine-tuning are the same as ECONET. The averages and standard deviations of Micro-F1 scores are reported from the runs with 3 different seeds. Since ECONET is the only model that targets both TORQUE and TB-Dense, we compare our results with it.

Our method achieves an F1 score of 65.8% on this task, compared to a RoBERTa-large baseline that achieves an F1 score of 63.4%. Moreover, our method outperforms ECONET, which requires an external corpus unlike ours. These results demonstrate that TRN's ability to build and utilize a predicted timeline is effective at various temporal relation understanding tasks, and as such, our method

[6] Though one may argue adding the extraction step with *CL* may further improve *CL*, we found this was not the case (F1 and EM of 75.4 and 50.7 respectively), which is why we report CL itself.

[7] https://github.com/PlusLabNLP/ECONET

has broader applicability beyond TRC.

6 Conclusion

We introduce a novel approach for temporal relation understanding, which elicits inductive reasoning behaviors by predicting time spans of events. Specifically, TRN is collectively supervised from a span-based timeline built from multiple questions on the same event, as stronger evidence than answer overlaps that spuriously lead to point-wise timeline.

TRN consists of the evidence extraction step that extracts individual evidence by answering each question with syntactic and semantic features, and the evidence chaining step that performs inductive reasoning for timeline prediction through a novel attention mechanism. Results on TORQUE and TB-dense datasets demonstrate that TRN outperforms previous methods by effectively mitigating the spurious answer overlaps.

7 Limitations

Despite the promising results, there are some limitations to our approach. One limitation is that since we rely on the predicted timeline to mitigate spurious overlaps, it still has a chance of error. knowledge distillation or meta-learning could be applied in the future to remove the potential error. Another limitation is that our main target, temporal reading comprehension, while a more realistic setting, is not commonly encountered in current NLP tasks. However, we argue that this is an important area that needs more active research, especially considering applications of NLP models in real-world and real-time scenarios. Moreover, while our primary focus has been on advancing methods for temporal understanding, it is important to highlight that our approach extends beyond this specific domain such as logical and causal reasoning. These domains share a common thread of requiring inductive reasoning skills, demonstrating the applicability of our proposed method.

For potential risks, our approach does not pose any significant risks. However, we note that our work utilizes PLMs so biases may exist in the models due to the nature of their training data.

Acknowledgement

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) [NO.2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University)] and Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2022-0-00077, AI Technology Development for Commonsense Extraction, Reasoning, and Inference from Heterogeneous Data).

References

Taylor Cassidy, Bill McDowell, Nathanael Chambers, and Steven Bethard. 2014. An annotation framework for dense event ordering. In *Annual Meeting of the Association for Computational Linguistics*.

Chunkit Chan, Jiayang Cheng, Weiqi Wang, Yuxin Jiang, Tianqing Fang, Xin Liu, and Yangqiu Song. 2023. Chatgpt evaluation on sentence level relations: A focus on temporal, causal, and discourse relations.

Kunlong Chen, Weidi Xu, Xingyi Cheng, Zou Xiaochuan, Yuyu Zhang, Le Song, Taifeng Wang, Yuan Qi, and Wei Chu. 2020. Question directed graph attention network for numerical reasoning over text. In *Conference on Empirical Methods in Natural Language Processing*.

Wenhu Chen, Xinyi Wang, and William Yang Wang. 2021. A dataset for answering time-sensitive questions. *arXiv preprint arXiv:2108.06314*.

Fei Cheng and Yusuke Miyao. 2017. Classifying temporal relations by bidirectional lstm over dependency paths. In *Annual Meeting of the Association for Computational Linguistics*.

Tianqing Fang, Zhaowei Wang, Wenxuan Zhou, Hongming Zhang, Yangqiu Song, and Muhao Chen. 2023. Getting sick after seeing a doctor? diagnosing and mitigating knowledge conflicts in event temporal reasoning.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. 2020. Evaluating models' local decision boundaries via contrast sets. *Findings of Empirical Methods in Natural Language Processing*.

Rujun Han, Xiang Ren, and Nanyun Peng. 2021. ECONET: Effective continual pretraining of language models for event temporal reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5367–5380, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2022. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python.

Hao Huang, Xiubo Geng, Guodong Long, and Daxin Jiang. 2022. Understand before answer: Improve temporal reading comprehension via precise question understanding. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 375–384, Seattle, United States. Association for Computational Linguistics.

Yinya Huang, Meng Fang, Yu Cao, Liwei Wang, and Xiaodan Liang. 2021. Dagn: Discourse-aware graph network for logical reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5848–5855.

Thomas Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *ArXiv*, abs/1609.02907.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Puneet Mathur, Rajiv Jain, Franck Dernoncourt, Vlad Morariu, Quan Hung Tran, and Dinesh Manocha. 2021. TIMERS: Document-level temporal relation extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 524–533, Online. Association for Computational Linguistics.

Qiang Ning, Hao Wu, Rujun Han, Nanyun Peng, Matt Gardner, and Dan Roth. 2020. TORQUE: A reading comprehension dataset of temporal ordering questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1158–1172, Online. Association for Computational Linguistics.

Qiang Ning, Hao Wu, and Dan Roth. 2018. A multi-axis annotation scheme for event temporal relations. In *Annual Meeting of the Association for Computational Linguistics*.

Qiu Ran, Yankai Lin, Peng Li, Jie Zhou, and Zhiyuan Liu. 2019. NumNet: Machine reading comprehension with numerical reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2474–2484.

Chao Shang, Peng Qi, Guangtao Wang, Jing Huang, Youzheng Wu, and Bowen Zhou. 2021. Open temporal relation extraction for question answering. In *Conference on Automated Knowledge Base Construction*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio', and Yoshua Bengio. 2017. Graph attention networks. *ArXiv*, abs/1710.10903.

Shuaicheng Zhang, Qiang Ning, and Lifu Huang. 2022. Extracting temporal event relation with syntax-guided graph transformer. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 379–390, Seattle, United States. Association for Computational Linguistics.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3363–3369.

A ChatGPT evaluation

Table 6 presents the evaluation results of ChatGPT (gpt-3.5-turbo-0301) on the TORQUE dataset. The outcomes reveal that the performance of ChatGPT on the TRC task is significantly inferior to that of a fine-tuned model, consistent with the observation that ChatGPT performs poorly on TRE tasks (Chan et al., 2023).

Regarding the evaluation settings, we introduced prompts for TORQUE as elaborated in Table 7 and conducted in-context learning (ICT). Then the model outputs are separated by commas to obtain the answers.

B Qualitative analysis

Figure 5 shows the qualitative analysis that compares the predictions of TRN with contrastive learning.

Models	Dev			Test		
	F1	EM	C	F1	EM	C
Fine-tuned RoBERTa-large	75.7	50.4	36	75.2	51.1	34.5
ChatGPT ICT$_{1shot}$	28.2	26.1	2.7	30.4	28.4	3.5
ChatGPT ICT$_{3shot}$	33.5	29.4	3.5	34.3	30.9	3.5

Table 6: The performance of ChatGPT vs fine-tuned RoBERTa on TORQUE dataset.

P1. After **touring** Tanzanian capital Dar es Salaam Thursday and **meeting** with Kenyan police leaders Friday morning, the FBI chief also **said** that he is very satisfied with the close and effective **cooperation** among the FBI agents and the police in Kenya and Tanzania. The man who **hurled** a grenade at security guards at the U.S. embassy here seconds before the bomb **exploded** was positively **identified** Thursday as two more suspects -- one Arab , one Sudanese -- who had been **arrested**, Kenya 's national newspapers **reported** Friday .	
Q1. What events had started before the FBI chief toured the Tanzanian capital?	*CL*: cooperation, hurled, exploded, __ TRN: cooperation, hurled, exploded, identified, arrested
Q2. What events occured after the FBI chief toured the Tanzanian capital?	*CL*: meeting, said, reported TRN: said, reported
Q3. What events occured before the FBI chief toured the Tanzanian capital?	*CL*: hurled, exploded, __ TRN: hurled, exploded, arrested
Q4. What events occured during the same time that the FBI chief toured the Tanzanian capital?	*CL*: meeting, cooperation, exploded, identified TRN: meeting

Figure 5: Qualitative analysis of contrastive learning and TRN. Events in the passage are highlighted in bold. In answers, correct events are denoted in blue, and incorrect events are denoted in red. Missing events are underlined.

Strategies	Template input

1-shot	`Question:` What had started before a woman was trapped? `Select answer events from the passage. One event corresponds to exactly one word. If there are no events, select None.`
	`Passage:` Heavy snow is causing disruption to transport across the UK, with heavy rainfall bringing flooding to the south-west of England. Rescuers searching for a woman trapped in a landslide at her home in Looe, Cornwall, said they had found a body.
	`Answer:` snow, rainfall, landslide
	
	`Question:` What happened before something was not mentioned? `Select answer events from the passage. One event corresponds to exactly one word. If there are no events, select None.`
	`Passage:` Titled "Beyond Human", the script "threw in a lot about UFOs and space aliens and earthlings evolving from their 'containers' to a 'higher level,"' Papas said. Suicide is not mentioned in the script, he added.
	`Answer:`

3-shot	`Question:` What had started before a woman was trapped? `Select answer events from the passage. One event corresponds to exactly one word. If there are no events, select None.`
	`Passage:` Heavy snow is causing disruption to transport across the UK, with heavy rainfall bringing flooding to the south-west of England. Rescuers searching for a woman trapped in a landslide at her home in Looe, Cornwall, said they had found a body.
	`Answer:` snow, rainfall, landslide
	
	`Question:` What happened before Lisa Schlein reports? `Select answer events from the passage. One event corresponds to exactly one word. If there are no events, select None.`
	`Passage:` Special events are being organized by the European Commission and individual nations in Europe, North America, and other parts of the world. Lisa Schlein reports from Geneva.
	`Answer:` None
	
	`Question:` What could have happened after the votes were provided? `Select answer events from the passage. One event corresponds to exactly one word. If there are no events, select None.`
	Passage: But instead of providing the votes to strike it down, they chose to uphold it on the flimsy ground that because the sex of the parent and not the child made the difference under the law, the plaintiff did not have standing to bring the case. The Justice Department, which supported the statute, did not cover itself with glory either.
	`Answer:` strike, have, bring, cover
	
	`Question:` What happened before something was not mentioned? `Select answer events from the passage. One event corresponds to exactly one word. If there are no events, select None.`
	`Passage:` Titled "Beyond Human", the script "threw in a lot about UFOs and space aliens and earthlings evolving from their 'containers' to a 'higher level,"' Papas said. Suicide is not mentioned in the script, he added.
	`Answer:`

Table 7: ChatGPT prompt templates used for the TORQUE dataset.